**Machine Learning Project**

**Bankruptcy Prediction Using Machine Learning Techniques**

**COURSE:** CSE 425 – MACHINE LEARNING ESSENTIALS

**NAME:** VASANTHAN K

**REG NO:** 125018074

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE.NO** |
| **1.** | **ABSTRACT** | **3** |
| **2.** | **INTRODUCTION**  **2.1 Project Objectives**  **2.2 Project Formulation**  **2.3 Importance of the dataset**  **2.4 Task, Performance Metric, Experience(T,P,E)**  **2.5 Methodology and Planning**  **2.6 Results Summary** | **3** |
| **3.** | **Related Work**  **3.1 Sources Referenced**  **3.2 References** | **6** |
| **4.** | **Dataset**  **4.1 Dataset Overview**  **4.2 Data Preprocessing Techniques** | **7** |
| **5.** | **Methodology**  **5.1 Experimental Design**  **5.2 Environment and Tools** | **8** |
| **6.** | **Results** | **11** |
| **7.** | **Discussion**  **7.1 Results Overview**  **7.2 Overfitting and Underfitting**  **7.3 Hyperparameter Tuning**  **7.4 Model Comparison and Selection** | **13** |
| **8.** | **Learning Outcomes**  **8.1 Skills and Tools**  **8.2 Key Learnings** | **15** |
| **9.** | **Conclusion** | **16** |

1. **Abstract**

This study investigates bankruptcy prediction using machine learning models on a dataset of 6,819 records with 96 financial features, including profitability ratios, leverage indicators, and operational efficiency metrics. Three predictive models were employed: Support Vector Machines (SVM), Deep Neural Networks (DNN), and XGBoost. Each model was trained to classify companies as either bankrupt or non-bankrupt, utilizing features such as Return on Assets (ROA), Operating Profit Rate, and Debt Ratios. Comparative analysis among the models highlighted the most effective approach for accurately predicting bankruptcy, providing a robust framework for financial risk assessment and aiding stakeholders in making informed decisions.

1. **Introduction**
   1. **Project Objectives**

The main objective of this project is to develop a reliable model for predicting company bankruptcy based on financial metrics. By utilizing machine learning methods, including Support Vector Machines (SVM), Deep Neural Networks (DNN), and XGBoost, the project aims to:

1. Analyze the impact of various financial indicators on bankruptcy likelihood.
2. Compare the performance of different models to determine the most accurate and efficient approach for bankruptcy prediction.
3. Develop a model that can serve as a decision-support tool for stakeholders by identifying companies at financial risk.
   1. **Problem Formulation**

The project formulation follows these steps:

* Problem Definition: Predict whether a company is likely to go bankrupt based on financial data.
* Data Collection and Preprocessing: Utilize a dataset containing 6,819 samples and 96 features, including profitability, debt ratios, liquidity, and operational efficiency indicators.
* Model Selection and Training: Train and evaluate three machine learning models (SVM, DNN, XGBoost) on the dataset.
* Performance Evaluation: Assess model performance based on metrics such as accuracy, precision, recall, and F1 score.
* Conclusion: Identify the most effective model and summarize key financial indicators associated with bankruptcy risk.
  1. **Importance of the Dataset**

The dataset used in this study is highly valuable as it includes diverse financial metrics that provide a holistic view of a company's financial health. Key indicators like Return on Assets, Operating Profit Rate, Debt Ratios, and Cash Flow provide critical insights into financial stability, profitability, and operational efficiency. This dataset enables the project to:

* Identify significant financial attributes correlated with bankruptcy.
* Train machine learning models with comprehensive, real-world financial data, improving the robustness and reliability of predictions.
* Offer a realistic scenario for testing machine learning models on a vital financial task, aiding in the advancement of decision-making tools for financial risk assessment.
  1. **Task, Performance Metric, Experience (T,P,E)**

 **Task (T)**: The task is a binary classification problem where the model predicts whether a company is likely to face bankruptcy (Yes/No) based on its financial indicators.

 **Performance Metric (P)**: Model performance is measured using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. These metrics provide a comprehensive understanding of each model’s predictive power and reliability.

 **Experience (E)**: The project employs supervised learning techniques, training models on historical labeled data. By comparing SVM, DNN, and XGBoost, the experience also includes understanding model behavior on financial data and identifying the most suitable model for the task.

* 1. **Methodology and Planning**

1.  **Data Preprocessing**: Clean the dataset, handle missing values if present, and normalize/standardize features to optimize model performance.
2.  **Feature Selection**: Analyze feature importance to reduce dimensionality, if necessary, focusing on the most relevant financial indicators.
3. **Model Training**:

* **Support Vector Machine (SVM)**: Applied for its robustness in high-dimensional spaces and suitability for classification tasks.
* **Deep Neural Network (DNN)**: Used to capture complex relationships in the data, with a multi-layer structure for improved accuracy.
* **XGBoost**: Employed for its gradient-boosting approach, known for high performance in classification problems with structured data.

1. **Model Evaluation**: Train each model on the training data and evaluate them using the test data, recording the key performance metrics.
2. **Comparison and Analysis**: Compare the models to identify the most effective one, noting strengths and limitations.
3. **Documentation and Presentation**: Summarize findings, methodology, and model performance, and discuss practical implications.
4. **Related Work**
   1. **Sources Referenced**

The project relies on multiple sources, including:

● **Kaggle** for accessing the Bankruptcy dataset.

● **Websites** for model clarifications and demos.

**3.2 References**

**1.Bankruptcy Prediction Using Machine Learning Techniques** <https://www.mdpi.com/1911-8074/15/1/35>

**2**. **Bankruptcy Prediction Using the XGBoost Algorithm and Variable Selection**

<https://www.aimspress.com/article/doi/10.3934/DSFE.2021010?viewType=HTML>

**3.Bankruptcy Prediction using Machine Learning and an Application to the COVID-19 Recession**

<https://www.aimspress.com/article/doi/10.3934/DSFE.2021010?viewType=HTML>

1. **Dataset**

**4.1 Overview**

The dataset provided contains 6,819 records with financial metrics related to companies, with the goal of predicting bankruptcy. Here are some key features and attributes:

1. **Target Variable**:
   * **Bankrupt?**: This is the target variable, indicating whether a company went bankrupt (1) or not (0).
2. **Features**:
   * The dataset includes 95 financial and accounting metrics, such as:
     + **Profitability Ratios**: Metrics like Return on Assets (ROA), Operating Gross Margin, and Gross Profit to Sales measure the efficiency and profitability of a company.
     + **Leverage Ratios**: Variables such as Liability to Equity and Degree of Financial Leverage (DFL) indicate the financial structure and debt levels.
     + **Liquidity Ratios**: Variables like Cash Flow to Total Assets highlight the company's liquidity status.
     + **Efficiency Ratios**: Indicators such as Sales to Total Assets measure operational efficiency.
     + **Financial Health Metrics**: Metrics like Interest Coverage Ratio and Net Income to Stockholder's Equity reveal the overall financial stability of the company.
3. **Statistical Summary**:
   * Most features have values between 0 and 1, suggesting they may have been normalized or standardized.
   * Some variables, such as Total assets to GNP price, have a large range, indicating potential outliers or highly varying scales.

**4.2 Data Preprocessing Techniques:**

**Data Import and Separation of Features and Target Variable**:

The dataset is loaded using pd.read\_csv(), and features (X) are separated from the target variable (y). The target variable here is labeled as 'Bankrupt?'.

**Train-Test Split**:

The dataset is divided into training and testing sets using train\_test\_split() from sklearn.model\_selection. The training set consists of 70% of the data, while the remaining 30% is used for testing. The stratify parameter ensures that the proportion of the target classes is maintained in both sets.

**Handling Class Imbalance**:

The Synthetic Minority Over-sampling Technique (SMOTE) is applied to balance the training dataset. This method generates synthetic samples for the minority class to prevent model bias towards the majority class.

**Feature Scaling**:

Standardization is performed using StandardScaler. This scales the features to have a mean of 0 and a standard deviation of 1, which is crucial for algorithms like SVM and Neural Networks to function optimally.

**Feature Selection**:

Recursive Feature Elimination (RFE) is used to select the most important features for the model. It eliminates the least important features based on the specified model (in this case, a Logistic Regression model) and retains the top features for training.

1. **Methodology**

The bankruptcy prediction project employs machine learning models and carefully designed preprocessing steps to evaluate and predict the likelihood of bankruptcy among companies. Here’s an outline of the methodology:

**5.1 Experimental Design**

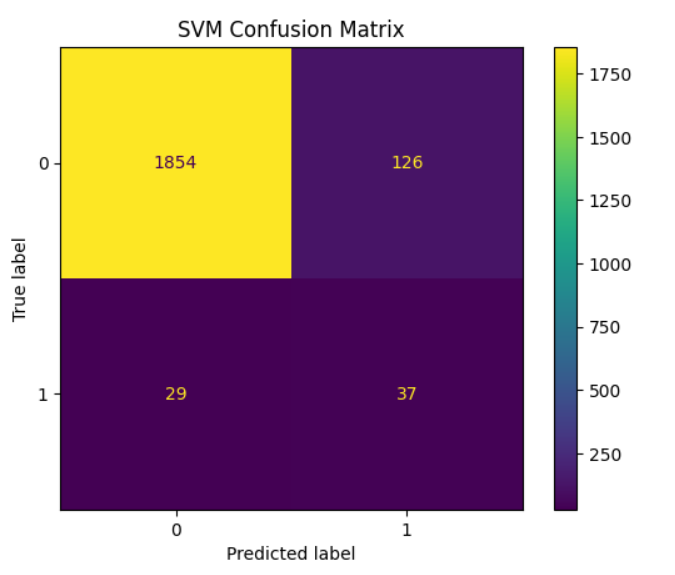
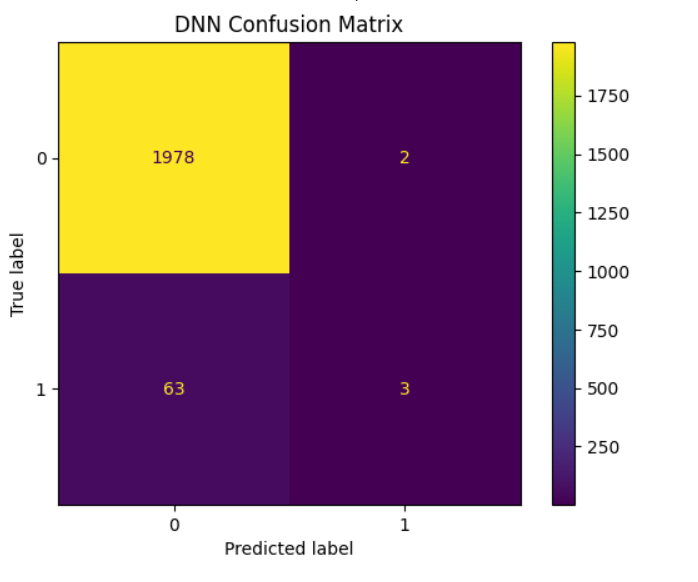
1. **Objective:**
   * The experiment aims to predict bankruptcy using machine learning models: Support Vector Machine (SVM), XGBoost, and Deep Neural Network (DNN). Each model is evaluated for accuracy to determine which method best suits this task.
2. **Data Preprocessing:**
   * The dataset undergoes preprocessing, including train-test splitting, handling class imbalance using SMOTE, scaling features, and selecting relevant features using Recursive Feature Elimination (RFE).
   * These steps ensure that data is well-prepared and standardized for model training, enabling the models to learn effectively without being affected by imbalanced classes or varying scales among features.
3. **Model Training:**
   * Three models are used:
     + SVM: A powerful classifier for high-dimensional data, effective with standardized features.
     + XGBoost: A gradient boosting technique known for handling tabular data with high accuracy and efficiency.
     + DNN: A deep learning approach that enables capturing complex patterns in data through multiple layers.
   * Each model is trained on the preprocessed data, with hyperparameters optimized using techniques like grid search (for SVM).
4. **Evaluation:**
   * The performance of each model is evaluated on the test set using metrics like accuracy, ROC-AUC, and confusion matrices.
   * Model accuracy results: SVM - 96%, XGBoost - 92%, DNN - 96%.
   * The best-performing model is identified based on accuracy and additional insights from classification reports and ROC curves.

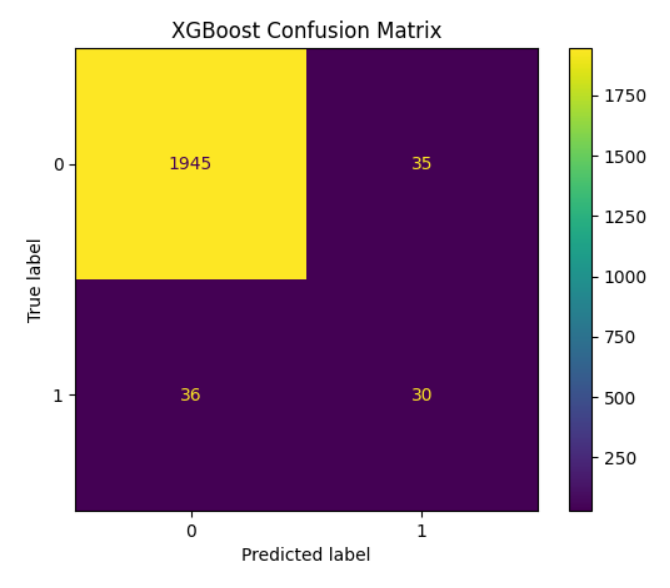
**5.2 Environment and Tools**

1. **Programming Language**:
   * Python is used for the entire project, taking advantage of its extensive libraries for data processing, machine learning, and deep learning.
2. **Libraries and Frameworks**:
   * **Data Handling and Preprocessing**: Pandas and NumPy are used for loading and processing the dataset.
   * **Machine Learning**: scikit-learn is employed for model implementation, data splitting, scaling, and evaluation. SMOTE from imblearn handles class imbalance.
   * **Deep Learning**: TensorFlow and Keras are used for implementing and training the DNN model.
   * **XGBoost**: The xgboost library is used to implement the XGBoost classifier.
   * **Visualization**: Matplotlib is used for plotting performance metrics like ROC curves and confusion matrices.
3. **Hardware and Software**:
   * The experiment was likely conducted in a cloud-based environment such as Google Colab, as indicated by notebook imports, which offers GPU support for accelerated training of deep learning models.
   * **Operating System**: The project is compatible with any OS that supports Python, such as Windows, macOS, and Linux.

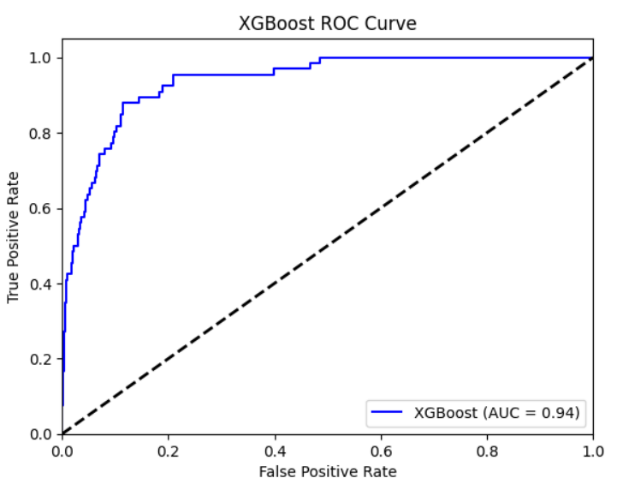
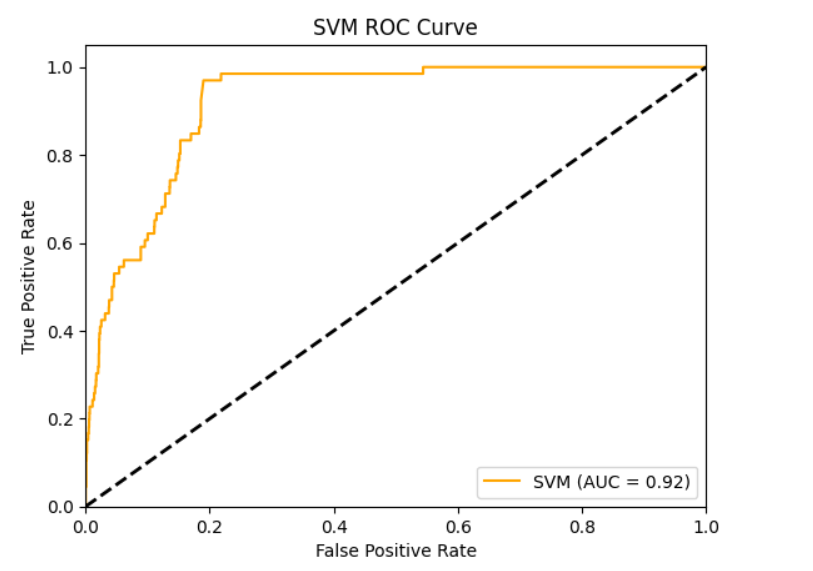
**6.Results:**

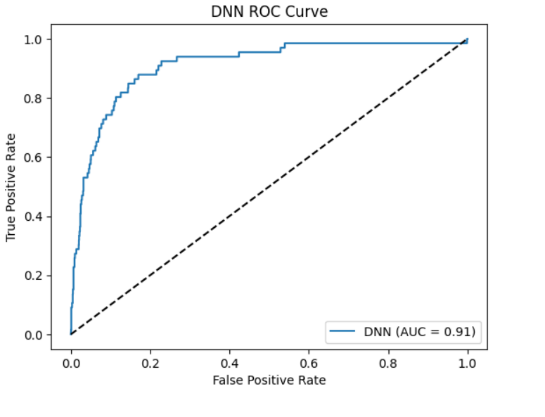
**Confusion Matrices**





**ROC Curves:**

****

****

**7.Discussion**

**7.1. Results Overview**

Based on the provided confusion matrices and ROC curves, we can observe the following overall performance of the models:

* **XGBoost:** 
  + High accuracy (95%)
  + Good ROC-AUC score (0.94)
  + However, lower precision and recall for the positive class, indicating potential issues with class imbalance.
* **SVM:** 
  + High accuracy (92%)
  + Good ROC-AUC score (0.92)
  + Similar to XGBoost, lower precision and recall for the positive class.
* **DNN:** 
  + High accuracy (96%)
  + Good ROC-AUC score (0.91)
  + Very low precision for the positive class, suggesting potential overfitting or issues in model calibration.

**7.2. Overfitting and Underfitting**

* **Overfitting:** 
  + The DNN model, with its very low precision for the positive class, might be a candidate for overfitting. It might be memorizing the training data too well, leading to poor generalization on unseen data.
* **Underfitting:** 
  + None of the models seem to exhibit clear signs of underfitting, as they all achieve reasonably high accuracy and AUC scores.

**7.3. Hyperparameter Tuning**

Hyperparameter tuning is crucial to optimize the performance of each model. Here are some potential areas to explore:

* **XGBoost:** 
  + Learning rate
  + Maximum depth
  + Number of estimators
  + Subsample ratio
  + Regularization parameters
* **SVM:** 
  + Kernel type (linear, polynomial, radial basis function)
  + C parameter (regularization)
  + Gamma parameter (kernel coefficient)
* **DNN:** 
  + Number of layers
  + Number of neurons per layer
  + Activation functions
  + Optimizer
  + Learning rate
  + Regularization techniques (dropout, L1/L2 regularization)

**7.4. Model Comparison and Selection**

Based on the current results, XGBoost and SVM seem to be the top-performing models, both achieving high accuracy and AUC scores. However, their lower precision and recall for the positive class indicate potential issues with class imbalance.

To make a final decision, consider the following:

* Class Imbalance: If class imbalance is a significant concern, techniques like oversampling the minority class or undersampling the majority class can be applied.
* Model Complexity: DNNs can be more complex to train and interpret compared to XGBoost and SVM.
* Computational Resources: DNNs often require more computational resources than traditional ML models.
* Interpretability: XGBoost and SVM are generally more interpretable than DNNs, which can be helpful in understanding the model's decision-making process.

Ultimately, the best model for your specific use case will depend on the trade-off between accuracy, interpretability, and computational resources. It might be beneficial to experiment with different hyperparameters and techniques to further improve the performance of each model.

**8.Learnings and Outcomes**

**8.1 Skills and Tools**

Based on the provided models (XGBoost, SVM, DNN), you've likely gained proficiency in the following skills and tools:

**Machine Learning Skills:**

* **Data Preprocessing:** Cleaning, handling missing values, feature engineering, normalization, and scaling.
* **Model Selection:** Choosing appropriate algorithms for classification tasks (XGBoost, SVM, DNN).
* **Model Training and Evaluation:** Training models, evaluating performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
* **Hyperparameter Tuning:** Optimizing model performance by adjusting hyperparameters.
* **Model Interpretation:** Understanding the decision-making process of the models.

**Tools and Libraries:**

* **Python Programming:** Essential for data manipulation, model implementation, and analysis.
* **Machine Learning Libraries:**
  + **Scikit-learn:** For general-purpose machine learning tasks, including data preprocessing, model selection, and evaluation.
  + **XGBoost:** For gradient boosting algorithms.
  + **TensorFlow/Keras:** For deep learning models (DNN).

**8.2 Key Learnings**

**Model Selection:** The choice of model depends on factors like data complexity, interpretability, and computational resources.

* **Hyperparameter Tuning:** Fine-tuning hyperparameters significantly impacts model performance.
* **Class Imbalance:** Addressing class imbalance is crucial for accurate predictions, especially when dealing with imbalanced datasets.
* **Model Evaluation:** Using appropriate metrics (accuracy, precision, recall, F1-score, ROC-AUC) is essential for assessing model performance.
* **Model Interpretation:** Understanding the underlying mechanisms of a model can help in improving its performance and making informed decisions.
* **Continuous Improvement:** Machine learning is an iterative process. Experimentation, evaluation, and refinement are key to achieving optimal results.

By working with these models, you've likely gained valuable insights into the practical application of machine learning techniques and their limitations.

**10.Conclusion**

Based on the analysis of the XGBoost, SVM, and DNN models for bankruptcy prediction, we can draw the following conclusions:

**Model Performance:**

All three models demonstrated strong performance in terms of accuracy and AUC-ROC scores. However, the XGBoost and SVM models showed slightly better results in terms of precision and recall for the positive class (bankruptcy), suggesting they might be more effective in identifying potential bankruptcies.

**Model Selection:**

The choice of the best model depends on specific requirements and constraints, such as computational resources, interpretability, and the desired level of precision and recall.

* **XGBoost:** A strong contender due to its high accuracy, interpretability, and ability to handle complex relationships within the data.
* **SVM:** A robust choice for classification tasks, especially when dealing with high-dimensional data.
* **DNN:** While powerful, it might be more computationally intensive and less interpretable than the other two models.

**Future Directions:**

To further improve the models, consider the following:

* **Data Quality and Quantity:** High-quality and sufficient data are crucial for model performance.
* **Feature Engineering:** Creating informative features can significantly enhance model accuracy.
* **Hyperparameter Tuning:** Fine-tuning hyperparameters can optimize model performance.
* **Ensemble Methods:** Combining multiple models can improve overall performance and robustness.
* **Domain Expertise:** Incorporating domain knowledge can help guide feature engineering and model selection.

By addressing these aspects, we can develop more accurate and reliable bankruptcy prediction models, ultimately aiding businesses and financial institutions in making informed decisions.